Proceedings of the First PhD Symposium on Sustainable Ultrascale Computing Systems (NESUS PhD 2016)
Timisoara, Romania

Jesus Carretero, Javier Garcia Blas
Dana Petcu
(Editors)

February 8-11, 2016

Beamforming filtering with real-time constraints on mobile embedded devices

Fran J Alventosa[1], Pedro Alonso[1], Gemma Piñero[2] and Antonio M Vidal[1]

[1]Dpto. de Sistemas Informáticos y Computación (DSIC)
[2]Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM)
Universitat Politècnica de València, Spain
{[1]fraalrue,[1]palonso,[1]avidal}@dsic.upv.es
[2]gpinyero@iteam.upv.es

Abstract

Nowadays Tables and Smart phones are equipped with low power processor. Some of them, like the NVIDIA Tegra SoC, also come with a GPU integrated so that both, the CPU and the GPU have access directly to the same RAM memory. In another vein, one the main limitations of microphone array algorithms for audio processing is the high computational cost required to reproduce real acoustics environments when real-time signal processing is absolutely required. One of these algorithms is the Beamforming Algorithm*, which is used to recover acoustic signals from their observations when they are corrupted by noise, reverberation and other interfering signals. In order to achieve real-time processing executing this algorithm we have employed high performance libraries such as OPENBLAS, LAPACK, CUBLAS, PLASMA and MAGMA, and a particular tune programming for these mobile devices.*

Keywords Heterogeneous Computing, Low Power Processors, ARMv7 and ARM Cortex-A15, Beamforming Filter

I. Motivation

The field of High-Performance Computing (HPC) has always been oriented to achieve good performance in terms of execution time. For this reason research in HPC has traditionally focused on applications of large computational cost on computers equipped with high-performance processors capable of performing large amounts of floating-point operations. Also on software tools and hardware resources addressed to large clusters of computers capable of working with large amounts of data. However, also in the field of high performance computing has always existed another type of needs represented by applications that, while not requiring the processing of a large amount of data (such as simulations), they do need immediacy in obtaining the result (real-time), as for example, a large set of applications of digital signal processing. It is also important to emphasize that we are experiencing a fundamental change in the conception of the Information and Communication Technologies ICT, moving from an oriented approach to the optimization of computational power and speed processes and applications to another approach more oriented to achieve maximum performance benefits at a low energy efficiency cost. This model change requires a new orientation in which efforts should be focused on the sustainability of the developments to ensure the optimum use of resources. The processor manufacturers are aware of this fact and design new devices that offer not only high computational performance but also a low consumption. For instance, the NVIDIA company delivers their graphics cards as devices of a high ratio Gflops per watt [1]. The ARM [2] is another example of processor that needs low energy to operate since it has been designed to be the core of mobile devices and, therefore, should be aware of the consumption to get the maximum availability.

II. Related work

There are many problems in engineering that can benefit from the good ratio of computational power by energy consumption offered by current processor architectures. The research group in which this doctoral thesis is integrated has a large experience in the design of high performance algorithms that address problems like 3D audio [3, 4], design of passive components based on microwave and electromagnetic devices applied to telecommunications [5, 6], systems analysis of detection of Multiple-Input Multiple-Output (MIMO systems) [7, 8, 9, 10], etc.

Typical paradigms of signal processing (detection, location, source tracking, feature extraction, etc.) have taken an extensive development in recent years in the form of distributed processed signals partly because of the increase of applications that have emerged around wireless sensor networks or, to be more specific, "Smart Sensors Networks" (*SSN*) obtained when the nodes of the network have processing and "decision making" capacities.

III. Thesis idea

The main target of this thesis is the design and implementation of algorithms for digital signal processing of sound signals in mobile devices. In an early step, we have tested the behaviour of high performance libraries of such HPC like BLAS [11], LAPACK [12], CUBLAS [13], PLASMA [14], and MAGMA [15], on an embedded system to evaluate their usability to solve our problem since many of the operations on which the algorithms are based can be cast in terms of linear algebra functions. We also have used parallel programming standards like OpenMP [16] and MPI [17].

The applications that can benefit from the work of this thesis are, e.g. applications of spatial sound (3D audio), filtering multichannel, echo cancellers of crosstalk, tracking and tracing of sources, classification and signal enhancement, etc. Among the applications, we will focus on processing distributed and collaborative signals around SSN's. Due to the high computational requirements to achive real-time processing we will try to get the best of the promissing NVIDIA solution SDK Jetson DevKit [18].

IV. The Beamforming Algorithm

In this section we make a brief introduction to the work being carried out in the framework of the thesis. This work consists in the effcient implementation of the Beamformer algorithm for the Jetson TK1.

Let $s_m(k)$, $m = 1, \ldots, M$, be signals emitted by M loudspeakers, the goal is to develop N filters g_n, $n = 1, \ldots, N$, where N is the number of microphones in the system, that allow to rebuild the original signals once cleaned from noise and room reverberation. To this end, we use channel responses of the room, represented as h_{nm}, for values of n and m stated before.

The output of the n-th microphone is given by:

$$x_n(k) = \sum_{m=1}^{M} \sum_{j=1}^{L_h} h_{nm}(j) s_m(k-j) + v_n(k) \ .$$

where L_h is the length of longest room impulse response of all the acoustic channels h_{nm}, and $v_n(k)$ is the noise signal. (For the sake of clarity, we will not consider the noise term hereafter.) Also for clarity and computation efficiency, we rewrite the form of the output signal of each microphone as

$$x_n(k) = \sum_{m=1}^{M} \mathbf{h}_{nm}^T \mathbf{s}_m(k) \ ,$$

where $\mathbf{s}_m(k)$ is the column vector defined as

$$\mathbf{s}_m(k) = \left[\ s_m(k) \quad s_m(k-1) \quad \cdots \quad s_m(k-L_h+1) \ \right]^T \ ,$$

and $\mathbf{h}_{nm}$ is the $\mathbb{R}^{L_h \times 1}$ acoustic channel vector from loudspeaker m to microphone n.

Considering now the problem of recovering source signals $s_m(k)$ from the recorded observations $x_n(k)$, beamforming filters g_n have to be designed so that the output signal $y(k)$ is a good estimate of $s_m(k)$, that is, $y(k) = \hat{s}_m(k-\tau)$ with minimum error. Given a maximum length of L_g taps for each of the N filters g_n, the broadband beamforming output signal is expressed in a similar form as

$$y(k) = \sum_{n=1}^{N} \mathbf{g}_n^T \mathbf{x}_n(k) \ ,$$

where $\mathbf{g}_n$ is the $\mathbb{R}^{L_g \times 1}$ vector containing the ordered taps of beamforming filters g_n, and $\mathbf{x}_n(k) = [x_n(k) x_(k-1) \cdots x_n(k-L_g+1)]^T$.

The algorithm of Beamformer filter called LCMV (Linearly Constrained Minimum Variance) [19] calculates beamforming filters as:

$$\mathbf{g}^{\mathrm{LCMV}} = \hat{\mathbf{R}}_x^{-1}\mathbf{H}_{:m}[\mathbf{H}_{:m}^T\hat{\mathbf{R}}_x^{-1}\mathbf{H}_{:m}]^{-1}\mathbf{u}_m\ , \qquad (1)$$

where $\mathbf{g}^{\mathrm{LCMV}}$ is formed by the concatenation of filters $\mathbf{g}_n$, i.e. $\mathbf{g}^{\mathrm{LCMV}} = [\mathbf{g}_1^T,\ldots,\mathbf{g}_N^T]^T$, and matrix $\mathbf{H}_{:m}^{(NL_g)\times(L_g+L_h-1)}$ is a partition of the channel impulse matrix that only includes the impulse responses from the m-th source to the N microphones used in *Sylvester* matrix form. Matrix $\hat{\mathbf{R}}_x$ is the correlation matrix of the recorded signals and $\mathbf{u}_m$ is the vector of zeros except for a one at the proper vector component in order to compensate the room impulse response delay.

The implementation of the LCMV proposed seeks for efficiency and accuracy, and its mainly based on the QR decomposition. Firstly, we form the following matrix $\mathbf{X} \in \mathbb{R}^{NL_g\times K}$,

$$\mathbf{X} = \frac{1}{\sqrt{K}}\begin{pmatrix} \mathbf{x}_1(k) & \mathbf{x}_1(k+1) & \ldots & \mathbf{x}_1(k+K-1) \\ \mathbf{x}_2(k) & \mathbf{x}_2(k+1) & \ldots & \mathbf{x}_2(k+K-1) \\ \vdots & \vdots & & \vdots \\ \mathbf{x}_N(k) & \mathbf{x}_N(k+1) & \ldots & \mathbf{x}_N(k+K-1) \end{pmatrix}, \qquad (2)$$

where $K\ (> NL_g)$ is the number of samples used. The algorithm computes the **qr** decomposition of $\mathbf{X}^T$, i.e. $\mathbf{X}^T = \mathbf{QR}$, where $\mathbf{Q}$ is orthogonal and $\mathbf{R}$ is upper triangular. Thus, in order to use LAPACK routines we build directly matrix X^T in column major order representation. Using matrix $\mathbf{X}$, matrix $\hat{\mathbf{R}}_x$ can be defined as

$$\hat{\mathbf{R}}_x = \mathbf{X}\mathbf{X}^T = \mathbf{R}^T\mathbf{Q}^T\mathbf{Q}\mathbf{R} = \mathbf{R}^T\mathbf{R}\,.$$

Now, we define for convenience matrix $\mathbf{W} = \hat{\mathbf{R}}_x^{-1}\mathbf{H}_{:m}$ so that the LCMV beamformer filter $\mathbf{g}^{\mathrm{LCMV}}$ (1) can be expressed as

$$\mathbf{g}^{\mathrm{LCMV}} = \mathbf{W}[\mathbf{H}_{:m}^T\mathbf{W}]^{-1}\mathbf{u}_m\ . \qquad (3)$$

We define matrix $\mathbf{Z}$ as the solution of the linear system

$$\mathbf{R}^T\mathbf{Z} = \mathbf{H}_{:m}\ ,$$

then, using the **qr** decomposition of matrix $\mathbf{X}$ we have

$$\mathbf{W} = \hat{\mathbf{R}}_x^{-1}\mathbf{H}_{:m} = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{H}_{:m} = \mathbf{R}^{-1}\mathbf{R}^{-T}\mathbf{H}_{:m} = \mathbf{R}^{-1}\mathbf{Z}\ ,$$

where clearly matrix $\mathbf{W}$ is the solution of the linear system $\mathbf{RW} = \mathbf{Z}$.

The solution to get the beamforming filters proceeds by solving the linear system

$$\mathbf{A}\mathbf{b}_m = \mathbf{u}_m\ , \qquad (4)$$

where $\mathbf{A} = \mathbf{H}_{:m}^T\mathbf{W} = \mathbf{H}_{:m}^T\mathbf{R}^{-1}\mathbf{Z} = \mathbf{Z}^T\mathbf{Z}$. Also here, the solution of the linear system (4) is obtained through a **qr** factorization, in this case, of matrix $\mathbf{Z}$. Let $\mathbf{Z} = \mathbf{Q}'\mathbf{R}'$ be the **qr** decomposition of matrix $\mathbf{Z}$, then vector $\mathbf{b}_m$ can be computed by solving the following two triangular linear systems:

$$\begin{aligned} \mathbf{R}'^T\mathbf{y} &= \mathbf{u}_m\ , \\ \mathbf{R}'\mathbf{b}_m &= \mathbf{y}\ . \end{aligned}$$

Finally, it is easy to see that the computation of the beamformer filter (1) can be computed using the last obtained objects, i.e. $\mathbf{R}$, $\mathbf{Z}$, and $\mathbf{b}_m$, this way:

$$\mathbf{g}^{\mathrm{LCMV}} = \mathbf{R}^{-1}\mathbf{Z}\mathbf{b}_m\ ,$$

which involves a matrix vector product and a triangular linear system solution.

The results have been carried out on the NVIDIA Jetson TK1, which consists of an ARM cortex A-15 with four cores and an NVIDIA GPU Kepler with 192 cores integrated all together in a single chip. The cost of the QR decomposition of matrix X (2) is $\approx 70\%$ the total cost of the algorithm, thus we focused our efforts on optimizing this operation. For the reduction in time of the QR decomposition we wrote different implementations based on libraries BLAS and LAPACK. After some testing we selected the optimized BLAS implementation OPENBLAS for the architecture ARMV7 as the best. We also used CUBLAS, PLASMA and MAGMA libraries to involve the GPU in the computations and, thus, to reduce the execution time.

In a first assessment we realize that MAGMA library is not (yet) optimized for devices with the characteristics of the Jetson (CPU and GPU ensambled on a single chip), since the cost of the QR decomposition by MAGMA is higher than the cost of our own implementation of the QR decomposition. Our implementation uses the same scheme as function GEQRF of LAPACK, but some operations are deliverd to the ARM processor cores using OPENBLAS and other operations are driven to the GPU using the CUBLAS library.

V. Conclusion and future work

Probably, the main conclusion of our incipient work is that yet exists a large room for improvement, both in the hardware devices as in the implementations that can exploit these devices. One of the solutions in which we are working on now consists of the QR updating. With this idea, many operations involved in the original algorithm that computes the QR factorization from scratch at each iteration can be avoided, allowing thus to reduce significantly the execution time.

References

[1] NVIDIA JETSON TK1, http://blogs.nvidia.com/blog/2013/11/20/10-greenest-powered-by-nvidia-gpus/, (accessed 2016 January 13).

[2] ARM Processors, http://www.arm.com/products/processors/, (accessed 2016 January 13).

[3] J. A. Belloch, M. Ferrer, A. González, F. J. Martínez and A. M. Vidal, "Headphone-Based Virtual Spatialization of Sound with a GPU Accelerator" in *J. Audio Eng. Soc.*, vol. 61, no. 7/8, pp. 546-561, 2013.

[4] J. A. Belloch, A. González, F. J. Martínez and A. M. Vidal, "Multichannel Massive Audio Processing using GPU" in *Integrated Computer-Aided Engineering (ICAE)*, vol. 20, no. 2, pp. 169-182, 2013.

[5] A. M. Vidal, A. Vidal, V. E. Boria and V. M. García, "Parallel computation of arbitrarily shaped waveguide modes using BI-RME and Lanczos Methods" in *Communications in Numerical Methods in Engineering*, vol. 23, no. 4, pp. 273-284, 2007.

[6] V. M. García, A. Vidal, V. E. Boria and A. M. Vidal, "Efficient and accurate waveguide mode computation using BI-RME and Lanczos methods" in *International Journal for Numerical Methods in Engineering*, vol. 65, no. 11, pp. 1773-1788, 2006.

[7] C. Ramiro, A. M. Vidal, A. González and S. Roger, "MIMOPack: a high-performance computing library for MIMO communication systems" in *Journal of Supercomputing*, vol. 71, no. 2, pp. 751-760, 2014.

[8] C. Ramiro, M. A. Simarro, F. J. Martínez, A. M. Vidal and A. González, "A GPU implementation of an iterative receiver for energy saving MIMO ID-BICM systems" in *Journal of Supercomputing*, vol. 70, no. 2, pp. 541-551, 2014.

[9] V. M. García, A. M. Vidal, A. González and S. Roger, "Improved Maximum Likelihood detection through sphere decoding combined with box optimization" in *Signal Processing*, vol. 98, no. 1, pp. 284-294, 2014.

[10] S. Roger, C. Ramiro, A. González, V. Almenar and A. M. Vidal, "An Efficient GPU Implementation of Fixed-Complexity Sphere Decoders for MIMO Wireless Systems" in *Integrated Computer-Aided Engineering (ICAE)*, vol. 19, no. 4, pp. 341-350, 2012.

[11] BLAS Library, http://www.netlib.org/blas/, (accessed 2016 January 13).

[12] LAPACK Library, http://www.netlib.org/lapack/, (accessed 2016 January 13)..

[13] CUBLAS Library, http://docs.nvidia.com/cuda/cublas/, (accessed 2016 January 13)..

[14] PLASMA Library, http://icl.cs.utk.edu/plasma/, (accessed 2016 January 13).

[15] MAGMA Library, http://icl.cs.utk.edu/magma/, (accessed 2016 January 13).

[16] OpenMP, http://openmp.org/wp/, (accessed 2016 January 13).

[17] MPI, http://www.mpi-forum.org/, (accessed 2016 January 13).

[18] NVIDIA JETSON TK1, https://developer.nvidia.com/embedded/develop/hardware, (accessed 2016 January 13).

[19] Jorge Lorente, Gemma Piñero, Antonio M. Vidal, Jose Antonio Belloch, Alberto GonzÃąlez, "Parallel implementations of Beamforming desgign and filtering for microphone array applications," in *European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, August 2011, pp. 501-505.